

IDENTIFICATION OF INSECT DAMAGED WHEAT KERNELS USING TRANSMITTANCE IMAGES

Zehra Cataltepe[†], Enis Cetin[‡], Tom Pearson[§]

[†]Siemens Corp. Research Inc. 755 College Rd East, Princeton, NJ 08540

[‡]Bilkent University, Dept of E.E. Eng., TR-06533 Bilkent, Ankara, Turkey

[§]USDA, GMPRC, 1515 College Ave. Manhattan, KS, 66502

ABSTRACT

We used transmittance images and different learning algorithms to classify insect damaged and un-damaged wheat kernels. Using the histogram of the pixels of the wheat images as the feature, and the linear model as the learning algorithm, we achieved a False Positive Rate (1-specificity) of 0.2 at the True Positive Rate (sensitivity) of 0.8 and an Area Under the ROC Curve (AUC) of 0.86. Combining the linear model and a Radial Basis Function Network in a committee resulted in a FP Rate of 0.1 at the TP Rate of 0.8 and an AUC of 0.92.

1. INTRODUCTION

Infested wheat kernels cause loss of quality in the wheat products. They also cause a lot more damage if they are put into storage with other kernels. It is important to be able to identify insect damaged kernels so proper decisions can be made about them.

Current methods of insect detection such as cracking and flotation [1], infrared CO_2 analysis [2], immunological methods [3], NIR [4], and x-ray inspection [5] can be laborious, slow, expensive, and ineffective at distinguishing a sound kernel from a kernel that is internally infested. It is possible that the use of acoustics [6] to detect insects may serve as an alternative which would allow for recognition of kernels where the insect has already emerged as well as those in which the insect is still living inside the kernel. In this paper we describe a method to identify insect damaged kernels based on transmittance images. This method is fast and inexpensive compared with the other methods. Recently, reflection images of kernels have been used for identification of different types of grains [7].

We first segmented the individual wheat kernels from the original transmittance images. Then we used the histogram of pixel intensities from each kernel to decide if it

was insect damaged or not. We used a number of different algorithms, namely the linear model, quadratic model, k-nearest neighbor, linear model with weight decay and Radial Basis Function Network. Linear model was the best of all the algorithms with a False Positive Rate (1-specificity) of 0.2 at the True Positive Rate (sensitivity) of 0.8 and an Area Under the ROC Curve (AUC) of 0.86 ± 0.03 . Although the radial basis function network performed worse than the linear model (an AUC of 0.79 ± 0.05), a committee of a linear model and a radial basis function resulted in an improved FP Rate of 0.1 at the TP Rate of 0.8 and an AUC of 0.92. We also experimented with K-nearest neighbor model, quadratic model and linear model with weight decay (ridge regression). All of these learning methods resulted in worse performance than the linear model.

2. WHEAT IMAGES AND FEATURES

Hard red winter wheat (H2) was used to obtain the images. The insect damaged kernel images were taken from wheat infested with rice weevil and kept at about a moisture of 11%. Transmittance images were taken as 800 pixels/inch tif images using an Epson Expression 1680 scanner. The exposure was set to 20 and gamma to 1.22.

The original un-damaged and insect damaged wheat kernel images were taken all together in two different shots. First we segmented each single kernel out of the original pictures using the blue component of the RGB. We obtained 355 good and 364 insect damaged kernels. We rotated each image so they had the maximum height and minimum width. Please see figure 1 for some sample images. The background color was white, so we determined the borders of each wheat image based on the background color. The reflectance along the borders of the image were affecting the features, so we cropped 10 pixels from each pixel row on each side of the wheat.

The histogram of red component of the pixels colors over each wheat image was used as the input feature for the learning algorithm. The 256 different Red components were put into bins as follows: If the red value was less than

Email addresses: zehra.cataltepe@siemens.com,
cetin@ee.bilkent.edu.tr, tpearson@gmprrc.ksu.edu



Fig. 1. A sample of good and insect damaged kernel pictures.

or equal to 80 the pixel was added into bin 0. If it was larger than 250 it was added into the last bin. (Since there were almost no pixels with Red component 80, we chose the limit 80. We merged Red value 255 into the bin that contained 250–254.) Otherwise, the pixel was added into a bin in-between, each bin being responsible for 5 different red values resulting with a total of 36 input features. Since the bins with Red value less than 80 were almost always empty, we chose to put all pixels with a Red component of less than 80 into one bin. Since there would be only one Red value (255) in the last bin, we decided to add that to the bin for 250–254. Figure 2 shows the mean and standard deviation of features for all the available data. We assigned output 0 to the good kernels and 1 to the insect damaged kernels.

In addition to the histogram features, we tried two other features: the minimum, maximum and majority over 3x3 rectangles and the mean on the center of the wheat. We also tried using, in addition to the Red histogram, mean of Red, Green and Blue, hue, saturation, brightness and mean x and y of CIE_{xy}. However, the results didn't improve, so we don't report them here.

3. LEARNING ALGORITHMS

We used two exemplar-based algorithms: k-Nearest Neighbor and radial basis function (RBF) network [8], as well as two model based algorithms: linear and quadratic models. In order to see if regularization would help with the linear model, we also tried weight decay. The input features for all the algorithms were $\underline{x} \in \mathcal{R}^{36}$ and the corresponding outputs were $y \in \{-1, 1\}$. The inputs were normalized to have

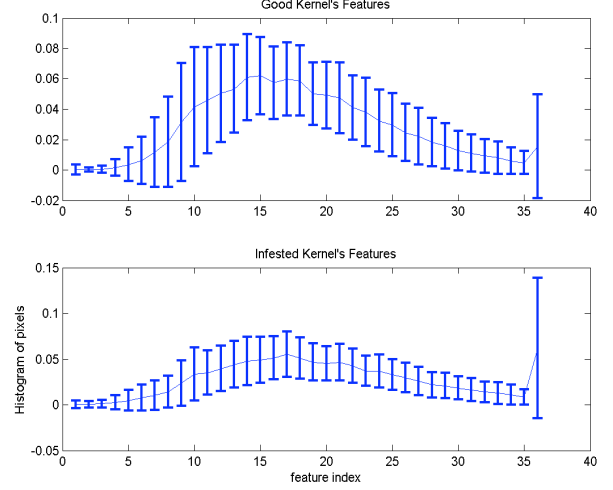


Fig. 2. The mean and standard deviation of the input features for good and insect damaged kernels.

sample mean 0 and standard deviation 1 for each input dimension on the training set.

In order to get reliable figures on algorithm performance, we used cross validation. We randomly partitioned all the available data into a training and a test set. The training set used 90% of data from each class and the test set used the remaining 10%. We repeated the partitioning 10 times.

We estimated the model performance using the ROC (Receiver Operating Characteristics) [9] and the Area Under the ROC curve (AUC) [10] on the test set. In order to obtain different False and True Positive rates on the ROC curve, we varied the threshold of each learning algorithm.

- **Linear Model:** Let $A_{N \times (36+1)}$ contain training inputs preceded by 1 and $\underline{b}_{N \times 1}$ contain the outputs y_i for all the N training examples. The linear model is obtained by solving for $\underline{w}_{37 \times 1}$ in the equation $A\underline{w} = \underline{b}$. In order to solve this equation we need to invert $A^T A$. Since A was not full rank, $A^T A$ was not invertible. We used singular value decomposition [11] with $\epsilon = 0.001$.

If the output for a test case was smaller than a certain threshold we classified it as good and otherwise we classified it as insect damaged. Each threshold for the linear classifier corresponds to a point on the ROC curve (i.e. a certain FP and TP rate). In order to get different points on the ROC curve, we varied the threshold for the output from -2 to 2 in steps of 0.1. For a certain threshold t and for a certain input, if the output of the linear model was more than the threshold, the input was classified as insect damaged, otherwise it is classified as good. When we varied the

threshold between -2 to 2 we were able to draw the complete ROC curve, that starts at TP and FP rates of 0 and ends at TP and FP rates of 1.

- **Radial Basis Function (RBF) Network:** We used the RBF network described in [8] choosing the first layer weights step-wise as the training example with the worst training error. We used 20 basis units. RBF network's first layer does a non-linear transformation of the inputs and then the output is determined as a linear combination of the basis function outputs.

We used thresholds as in the linear model to get different ROC curve points.

- **Linear Model and RBF Network Committee:** We used a linear combination of the RBF network and the linear model outputs as the output of the committee and the same thresholds to get ROC curve points.
- **Quadratic Model:** We used the inputs used for the linear model and also the multiplication of each input with another input.

We used thresholds as in the linear model to get different ROC curve points.

- **k Nearest Neighbor:** This algorithm needs to store all training data. In order to classify a new data point, first the K closest data points (K neighbors) in training data are determined. The new data point is classified as positive or negative, based on the count of positive and negative count in the K neighbors.

The number K determines the smoothness of the k Nearest Neighbor classifier [8]. As K increases the classifier does a smoother interpolation. We used 5, 10, 15 and 20 as the values of K in our experiments.

In order to get different points in the ROC curve, we varied the threshold for the output from 0 to 1. We computed the mean of the labels of the K nearest neighbors. If the mean is less than the threshold, we classified a test case as good and otherwise as insect damaged.

- **Linear Model with Weight Decay** Weight decay, ridge regression and shrinkage aim at reducing the weights and hence obtaining simple models that do not overfit the training data. The weight decay solution is $w^* = (A^T A + \lambda I)^{-1} A^T y$. The selection of the weight decay parameter λ is very important. If λ is very small, the weight decay doesn't change the solution, if it is too large, the solution gets smaller in size at the expense of bad fit to the data.

We used thresholds as in the linear model to get different ROC curve points.

Algorithm	AUC
Linear	0.86 ± 0.03
RBF	0.79 ± 0.05
RBF and Linear Committee	0.92 ± 0.03
Quadratic	0.85 ± 0.05
5 Nearest Neighbor	0.55 ± 0.02
10 Nearest Neighbor	0.77 ± 0.04
15 Nearest Neighbor	0.79 ± 0.03
20 Nearest Neighbor	0.76 ± 0.03
Weight Decay $\lambda = 0.002$	0.86 ± 0.02
Weight Decay $\lambda = 0.003$	0.87 ± 0.02
Weight Decay $\lambda = 0.004$	0.84 ± 0.02

Table 1. Area Under ROC Curve (AUC) for Different Learning Algorithms

4. RESULTS

For each of the 10 training-test set partitioning of the available data, we used the training set to train the learning algorithm. We then used the test set to compute the ROC (Receiver Operating Characteristics) [12, 9, 13] curve for each partitioning.

We interpolated the ROC curve for each partitioning and reported the mean and standard deviation of the True Positive Rate (sensitivity) for each False Positive Rate (1-specificity) value for each learning algorithm [9]. The mean and the standard deviation on the ROC curve gives us a better idea on the performance of an algorithm. In order to get a reliable mean, we discarded the ROC curve with the maximum and minimum AUC and computed the average ROC curve using the 8 remaining ROC curves. Please see table 1 and figure 3.

Because of its simplicity and performance linear model seems to be the best single algorithm. The nearest neighbor was the worst algorithm, regardless of the K of the nearest neighbor. The RBF and linear model committee performed the best.

5. DISCUSSION

We used a number of learning algorithms to classify good and insect damaged wheat kernels and we found out that the regularized linear model performed the best. Additional information about the kernels such as reflectance images or as compression force or conductance measurements [6] could be used to improve performance of a single classifier. Another approach is to train different classifiers with each of these features and then combining them [14].

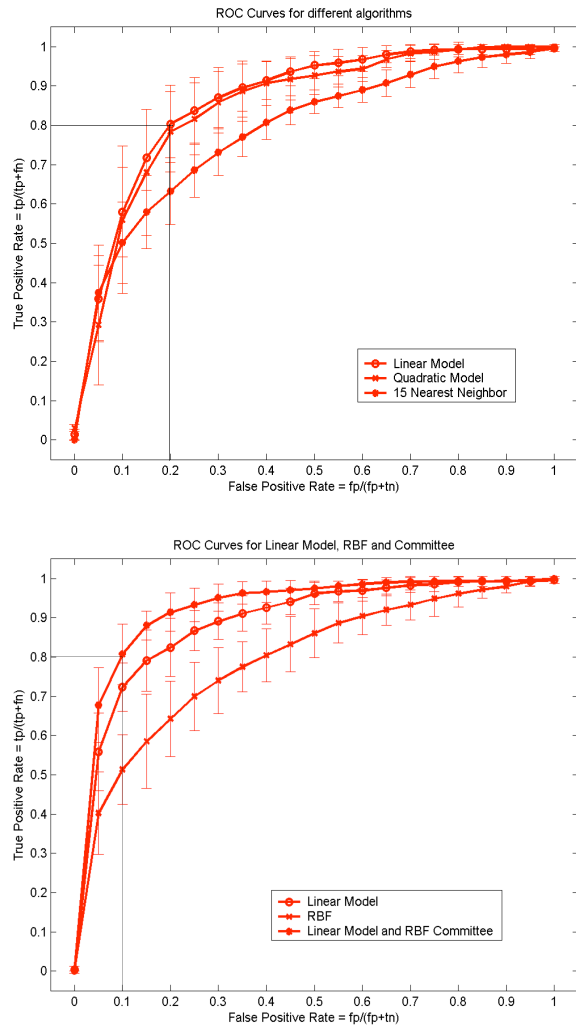


Fig. 3. Performance of Different Learning Algorithms.

6. REFERENCES

- [1] G.E. Russell, "Evaluation of four analytical methods to detect weevils in wheat: granary weevil, sitophilus granarius in soft white wheat," *Journal of Food Protection*, vol. 51, pp. 547–553, 1988.
- [2] W.A. Bruce, M.W. Street, R.C. Semper, and D. Fulk, "Detection of hidden insect infestations in wheat by infrared carbon dioxide gas analysis," *ARS bulletin*, vol. July, AAT-S-26, 1982.
- [3] F.A. Quinn, W. Burkholder, and G.B. Kitto, "Immunological technique for measuring insect contamination of grain," *Journal of Economic Entomology*, vol. 85(4), pp. 1463–1470, 1992.
- [4] F.E. Dowell, J.E. Throne, and J.E. Baker, "Automated nondestructive detection of internal insect infestation of wheat kernels by using near-infrared spectroscopy," *Journal of Economic Entomology*, vol. 91(4), pp. 899–904, 1998.
- [5] R.A. Stermer, "Automated x-ray inspection of grain for insect infestation," *Transactions of the ASAE*, vol. 15, pp. 1081–1085, 1972.
- [6] T. Pearson, D.L. Brabec, and C.R. Schwartz, "Automated detection of internal insect infestations in whole wheat kernels using a pertens skcs 4100," *Applied Engineering in Agriculture*, vol. 19(6), pp. 727–733, 2003.
- [7] N.S. Visen, D.S. Jayas, J. Paliwal, and N.D.G. White, "Comparison of two neural network architectures for classification of singulated cereal grains," *Canadian BioSystems Engineering*, vol. 46, pp. 3.7–3.13, 2004.
- [8] C. M. Bishop, *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford, 1995.
- [9] F. Provost, T. Fawcett, and R. Kohavi, "The case against accuracy estimation for comparing induction algorithms," in *Proc. 15th International Conf. on Machine Learning*. 1998, pp. 445–453, Morgan Kaufmann.
- [10] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, No. 7, pp. 1145–1159, 1997.
- [11] W. H. Press, W. T. Vetterling, B. P. Flannery, and S. A. Teukolsky, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, 1992.
- [12] C. E. Metz, "Basic principles of roc analysis," *Seminars in Nuclear Medicine*, vol. 8, pp. 283–298, 1978.
- [13] M. H. Zweig and G. Campbell, "Receiver-operating characteristic (roc) plots: a fundamental evaluation tool in clinical medicine," *Clinical Chemistry*, vol. 39, pp. 561–567, 1993.
- [14] J. Kittler M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20(3), pp. 226–239, 1998.